

# Conceptual design and modification of freeform surfaces using dual shape representations in augmented reality environments

Mark Fuge, Mehmet Ersin Yumer, Gunay Orbay, Levent Burak Kara \*

Visual Design and Engineering Laboratory, Carnegie Mellon University, United States

## ARTICLE INFO

### Keywords:

3D surface design  
Computer-aided design  
3D sketching  
Conceptual design  
Augmented reality  
Surface representations  
Hand motion  
Gesture

## ABSTRACT

This paper enables the rapid creation and modification of freeform surfaces inside an augmented reality environment, and focuses on methods for enabling increased flexibility during exploratory, conceptual industrial product design through three-dimensional (3D) sketch-based user input. Specifically, we address the role of multiple shape representations with varying uncertainty levels during 3D conceptual sketching, along with methods to transform between those representations. The main contributions of this work are: (1) the formulation of virtual shape data in multiple, concurrent representations (points and surfaces), and a regression method to transition fluidly back and forth between these representations during design, (2) methods for deforming and exploring the product shape using these multiple representations, and (3) representations of these forms such that designers can explore conceptual designs without the need for detailed surface operations such as trimming or continuity enforcement. Through incorporating these contributions, we introduce techniques that can be incorporated in future computer-aided conceptual design systems. These contributions are demonstrated for freeform surface design, with examples of computer mouse and car seat exterior surfaces.

© 2011 Elsevier Ltd. All rights reserved.

## 1. Introduction

Current computer-aided design (CAD) tools are well suited for detailed, downstream phases of the design process. However, CAD systems are currently of limited utility in early conceptual design [1,2]. This is due to complexity of their interfaces, which arise from (1) complex mathematical formulations to represent surfaces (NURBS, Beziér patches, etc.), (2) the time and expertise required to create and manipulate such geometries, and (3) the crisp and definite nature of the resulting geometries. Indeed, many studies have shown that overly concrete representations can be harmful to the early design process [3–7]. During conceptualization, the design is still developing and is imprecisely specified [1,8–10]. At this point in the design process, a designer is primarily concerned with the rapid generation and manipulation of ideas, preferring to defer unnecessary commitment and detailed design until later in the design process [11].

Based on these observations, prior research has studied the effects of conventional CAD systems on creativity and problem solving [11,12,4]. Robertson and Radcliffe, in particular, conclude that, while current CAD systems provide a means for enhanced visualization and communication, they also result in (1) circumscribed thinking, where design ideas are primarily

dictated by the capabilities of the software, (2) bounded ideation, where the software becomes a suboptimal environment for ideation, and (3) premature fixation, where detailed, crisp models convey an illusion of completeness that discourages designers from exploring alternatives. Future CAD tools need to take such issues into consideration [1,8,9,12].

As one step toward addressing some of these issues, we present a surface design method that uses natural human hand gestures to create and modify surface designs in an augmented reality (AR) environment. Our approach is motivated by the human ability to express ideas through hand gestures when describing shapes to one another. In many cases, humans seem to have little or no difficulty mentally reconstructing the described shapes, even though many of the details may be missing [13]. We believe that taking advantage of this natural way of describing shapes can mitigate some of the existing shortcomings of conventional CAD tools.

### 1.1. Freeform conceptual design and modification by dual shape representations

Many previous methods for conceptual design, such as sketch-based interfaces [14–17] or virtual reality based wireframe creation methods [18,19], require the designer to first construct all of the surface edges before being able to visualize the resulting surface. This detail-oriented process of edge construction can

\* Corresponding author. Tel.: +1 412 268 2509.  
E-mail address: [lkara@cmu.edu](mailto:lkara@cmu.edu) (L.B. Kara).

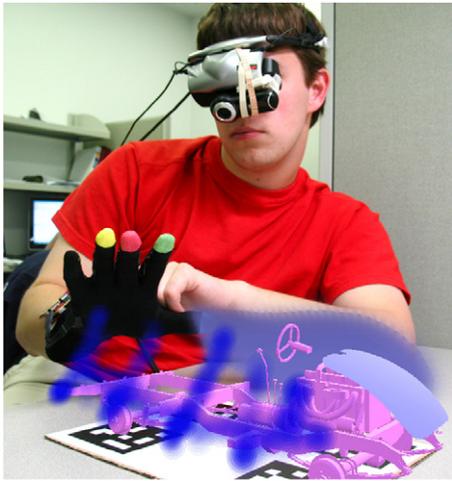


Fig. 1. Overview of the user interface.

impede rapid surface exploration—an aspect of current techniques that future CAD systems may not want to inherit.

In contrast, we present techniques for directly creating and modifying freeform surfaces through three-dimensional (3D) hand gestures, allowing designers to defer decisions such as edge boundary creation, surface trimming, etc. until after they have explored the overall shape. We posit that since people frequently perform similar hand gestures to each other when describing what a shape might look like, this may be an intuitive interaction paradigm for future CAD systems. To demonstrate these techniques, we have deployed them in the AR environment shown in Fig. 1.

Aside from interaction techniques, the choice of geometric representation critically affects the fluidity and interactivity of surface design exploration. On the one hand, allowing designers to represent geometry as a set of 3D point clouds provides them with the maximum flexibility in exploring a wide variety of shapes; they are not constrained by any specific parametric representation [1]. On the other hand, without any kind of underlying surface representation, CAD systems are not able to completely render a surface design for visual inspection—a critical feature if designers wish to visualize their surface design and decide whether or not it needs to be adjusted.

Given that CAD users may want different geometric representations of their surface design during different aspects of their exploration, our approach maintains a dual representation in which 3D points sampled from hand gestures can either be visualized as a point cloud or as a parametric surface. We allow designers to move back and forth between these two representations, affording them both the flexibility and visual feedback needed to iteratively explore the design space.

## 1.2. Contributions and significance

This work focuses on the following key points in the context of future computer-aided conceptual design tools.

1. Methods of surface creation and modification that utilize hand gestures as input rather than conventional input mechanisms (i.e. mouse and keyboard, or tablet interfaces).
2. Representing the geometry such that designers can explore conceptual designs without the need for detailed surface operations such as trimming or continuity enforcement. While these are important detailed design activities, they should not impede initial concept exploration.
3. The use of multiple shape representations, as well as techniques for moving between these multiple representations.

The first and second goals work toward enhancing the *visual representation and interpretation* mechanisms that are expected of future CAD tools [12], while moving the designer away from *bounded ideation* and *premature fixation* [11]. In the context of shape design, the last goal enables the first two, while maintaining geometric representations that facilitate exploration during the early stages of the design process.

The proposed approach provides virtual design systems with significant advantages over their low-fidelity sketching and clay modeling counterparts. These techniques facilitate iterative exploration and evaluation of low-fidelity conceptual surface designs prior to formal parametric CAD modeling. While an AR system is presented in this work, it should be noted that the primary goal of this paper is to demonstrate the foundations and utility of a new modeling approach, and as such does not include a usability evaluation of the AR system itself. This being said, the usage of an AR system in this work does play a critical role in how users interact with the design. By utilizing 3D optical tracking, our system can use this AR environment to transform user movements and gestures directly into 3D geometric data. This task is currently difficult to accomplish via mouse and keyboard or 2D sketch interfaces, and as such lends itself well to incorporation within an AR environment.

## 2. Related work

There are a vast number of prior approaches that have furthered the application of AR to computer-aided conceptual design. In order to give a coherent breakdown of this prior work, we organize the related literature around three topics. First, we review literature that has attempted to provide representations with uncertainty and ambiguity. Second, we review various approaches of moving between representations in CAD, and highlight the relevant challenges that remain. Finally, we review AR systems that have implemented techniques for manipulating and interacting with various representations.

### 2.1. Ambiguous shape representations for CAD

When tackling the problem of describing incomplete or ill-specified geometry, prior work has taken a number of approaches, including (1) applications of fuzzy set theory, (2) particle systems, and (3) probabilistic methods. A similar such delineation is presented in [20].

Starting in the early 1990s, Yamaguchi et al. introduced fuzzy set theory to CAD application through the idea of probabilistic solid modeling [2]. Fuzzy sets remain a popular choice for ambiguous data representation with applications to surface construction [21]. However, as mentioned by Martin in [22], various issues such as visualization, potential for interactivity, and common operations on geometry need to be refined before this approach can be adapted for AR product design systems.

By contrast, particle-based systems model geometries as sets of points, called particles, which can interact with one another. Introduced by Blinn [23] in 1982 for molecular models, particle systems are useful for expressing conceptual geometry as a set of points that can later be used to construct surfaces. More recently, the vague interval discrete modeling (VDIM) approach by Rusák and Horváth [24] serves as an example of using particle systems not just for visualization, but for conceptual design. The advantage of particle-based systems is that they provide a convenient representation for physically based modeling scenarios, which is desirable in virtual product design systems. However, the lone use of particle systems may not be conducive to direct surface creation and manipulation, if they are not integral parts of existing topology. Conversions from standalone particle representations to

surface-based definitions are not straightforward. The point-based representation in our work can be seen as a simplified particle system; however, we utilize a representation conversion scheme that allows our system to overcome the above-mentioned issue and preserve a dual representation structure throughout the entire conceptual design process.

Probabilistic models provide an alternative form, which considers statistical variations around a prescribed shape in order to provide shape ambiguity. An overview of some of these approaches, as well as an implementation for the design of 2D curves, is presented by Lim et al. in [25].

## 2.2. Representation conversion approaches

Representation conversion, more specifically, conversion from points to a surface, is a common issue found in point to surface reconstruction problems. A seminal work by Hoppe et al. [26] concerned the ability to reconstruct surfaces from unorganized point set data in the form of meshes, range images, and contours. The automobile industry uses surface reconstruction techniques extensively during conceptual design, and mesh growing techniques [27] also provide point to surface conversion.

Bio-modeling applications frequently convert between image representations to contours or surface representations, and bio-modeling has grown into an entire field in and of itself. A representative work whose techniques could be applied to our field is the work by Lelieveldt et al. [28], who use fuzzy sets to fit implicit surfaces for internal organ scans.

While there is a large body of literature dedicated to converting from one representation to another (points to surfaces, images to contours), most of this literature does not attempt to provide any bi-directional correspondence between these representations. For example, a change in a point set would alter the subsequent surface, but a modification to the surface does not uniquely impact the point set. This bi-directional mapping is a key element in using multiple representations during interactive design, and thus is a subsequent focus area in this work.

## 2.3. Interaction techniques in AR product design systems

Starting around 1991 with 3-Draw [29], there has been great interest in creating virtual or augmented reality product design systems. The number of prior systems is substantial, and Jimeno provides a review paper detailing many of them [30]. For purposes of comparison, we divide related systems by their primary choice of geometric representation: (1) point/stroke based, (2) surface based, and (3) volume based.

Point-based and stroke-based interactive modeling systems were some of the first systems to appear, due to the types of tracking technologies available (i.e. spatial point sampling devices). They remain popular due to user familiarity with pen-based interfaces. Prior approaches have used point representations in virtual tape drawing [31], in haptic feedback devices [32], and for rapid prototyping of concept forms [33]. In this area, the work most similar to ours is the use of 3D strokes and subsequent surface fitting algorithms by researchers at Fraunhofer IGD for automotive design [34,35]. However, purely pen-based interfaces limit creative exploration via physical deformations. In all of these systems the user must provide exact basis curves before surface exploration is possible. Dorman and Rockwood [36] introduce a vertex-based modification scheme which is similar to our point modification method; however, they directly alter mesh representations by pulling or pushing vertices affected by volumes of interest.

In contrast, surface-based creation methods are designed to allow the user to directly specify or modify the surface, without the need for prior strokes or constraints. Prior work has shown

the applicability of this representation for sweeping surface patches [19,37], haptic deformation [38], NURBS modeling [39], and ergonomic optimization [40]. Of the surface creation work, the system most similar to ours is “Surface Drawing” by Schkolne et al. [41]. This system allows the user to sweep their hand through the air, creating a surface based on hand curvature. The limitation of surface-based approaches is that expressing large patches requires the user to essentially “paint” the desired region. Our system uses an underlying fitted surface approach to circumvent the need for having to manually specify all visible surface polygons.

Several volume-based approaches have also been explored, primarily with respect to haptic feedback [42,43]. While volume-based systems tend to work well for many engineering applications where the geometry is expressed in geometric primitives, freeform surface design is substantially more complex, and current volume-based AR systems do not account for this complexity.

Together, all of the above work has provided a strong framework for manipulating geometry within their own particular chosen representation. Each representation showcases distinct advantages and limitations over the other. This work differentiates itself by allowing the user to utilize multiple representations, instead of attempting to improve just one. By providing this framework to the community, we hope that the capabilities of the above-mentioned systems can be utilized in complementary ways, provided that an effective representation conversion is achieved.

## 3. Technical approach

The following sections provide an overview of the problems and the proposed techniques motivated by our contributions presented in the introduction. Section 3.1 discusses the multiple representation concept in detail, providing two specific examples of point-based and surface-based representations. Section 3.2 highlights a specific method for converting back and forth between point-based and surface-based representations, providing the critical bi-directional link between the two. Finally, Section 3.3 describes methods for interacting with these shape representations in meaningful and intuitive ways based on deformation.

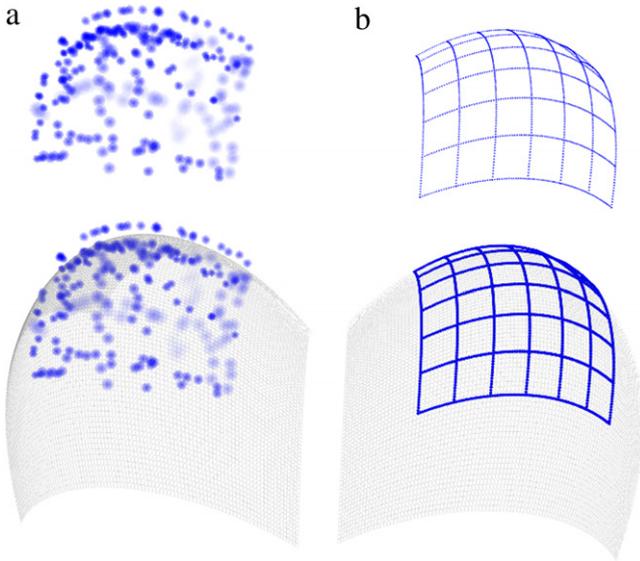
### 3.1. Dual point/surface representation

A key contribution of this work is the usage of multiple shape representations during the interactive design of a product form. This section expands on this idea by providing specific examples of how these representations can be implemented. Specifically, this work implements two representations: a point-based representation and an analytical surface-based representation based on polynomial surfaces.

#### 3.1.1. Point-based representation

Given only the instantaneous location of a user’s hand, the most fundamental representation is that of a single point in 3D space. However, only deciding to capture the point location itself neglects a wealth of other information regarding the user’s intentions. As in 2D sketching, designers tend to lightly define their ideas with faint strokes, before resketching over them in greater pressure and detail. This implies that designers choose to sketch with various degrees of certainty, often reflected in the pressure of the strokes.

To analogically capture this “degree of certainty” property for 3D points, the user uses his or her opposite hand to press on a force-sensitive pad on the back of the glove, which can record up to 1024 different levels of pressure. This creates a stream of single points located at the geometric center of all fingertips whose deposition rate is equal to the frame rate of the system. The pressure readings at the time of deposition define a confidence region around each point. In contrast to the points shown in 2(b), Fig. 2(a)



**Fig. 2.** Both these weighted least squares surfaces are identical. However, the point representation in (a) affords a greater degree of perceptual ambiguity than that in (b).

demonstrates how this weighting helps create more perceptually ambiguous renderings for points that have lower certainty. In this manner, the user is able to sweep his/her hand through any path in 3D space, and lay down points whose visual influence is controlled by the point's weight. The size and transparency of the rendered point is calculated as

$$R(\rho) = L \cdot (1 - \rho); T(r, \rho) = \rho \cdot \left(1 - \frac{r}{R(\rho)}\right), \quad (1)$$

where  $\rho$  is the recorded pressure, from 0 to 1,  $R(\rho)$  is the resulting point outer radius, and  $L$  is a maximum radius that is chosen based on the desired size scale. In all of the examples presented in the paper,  $L = 0.5$  inches. The transparency has a linear falloff with respect to the radius ( $r$ ) from  $\rho$  in the center to 0 when  $r = R(\rho)$ . In order to speed up rendering, our system discretizes this transparency falloff into five superimposed transparent spheres. This discretization can be seen by looking closely at each point in Fig. 6. This processing results in a continuum from certain points, which are small, well defined, and opaque to uncertain points, which appear larger, fuzzier, and more transparent.

### 3.1.2. Surface-based representation

In contrast, a surface-based representation defines an explicit or implicit surface using a functional equation. Unlike the abstractness of the point-based representation, the surface-based representation is definite in that it describes a potentially unbounded surface which the user can then modify. Ultimately, whether in a functional or mesh form, the final result of a conceptual design has to eventually become some well-defined representation of the target geometry. In our system, this surface-based representation provides that concreteness.

Although a surface-based representation can be any explicit or implicit function, our system deals with any  $N$ -order polynomial surface (cubic, quartic, etc.). The examples in the rest of the paper will consider mostly quadratic surfaces of the form

$$f(x, y) = a_0 + a_1x + a_2y + a_3xy + a_4x^2 + a_5y^2. \quad (2)$$

This is for simplicity of presentation and because cubic-order or lower polynomial surfaces were of sufficient complexity to capture the design intent of a range of different conceptual product designs presented in this paper. This polynomial surface representation is

calculated in a coordinate frame that is local with respect to an individual point cloud, instead of restricting all surfaces to one global coordinate frame. Further details of the construction of these surfaces are given in Section 3.3.3.

### 3.2. Converting between representations

It is the stark difference between the point-based and surface-based representations which makes their pairing valuable: the abstract point representation permits uncertainty and divergence, whereas a surface representation requires concreteness and convergence. However, in order to take advantage of both representations, we need to be able to easily transform between one representation and the other.

This *representation conversion* problem is the key challenge that needs to be overcome in order to bridge between design systems using different representations. Depending on the representations, creating this two-way conversion is not necessarily trivial. In the case of point to surface conversion, the tools of statistical regression help alleviate one direction. Through using a weighted least squares (WLS) approach, a given function  $f(\mathbf{x})$  can be found by taking a set of points and finding a set of weighted residuals such that Eq. (3) is minimized. In this case, the weighting function value  $\theta_i$  is given directly by the pressure value of the  $i$ th point.

$$\min \sum_i \theta_i \|f(\mathbf{x}_i) - f_i\|^2. \quad (3)$$

As an example, in the case of quadratic surfaces,  $\mathbf{x}_i$  are Cartesian coordinates  $x_i$  and  $y_i$ ,  $f_i = z_i$ , and  $f(\mathbf{x}_i)$  is expressed in Eq. (2). In this way, our system uses the Cartesian positions of preexisting points and performs a regression on them to determine the unknown coefficients in Eq. (2).

One of the main advantages of using regression to fit the surface representation is that our system can reconstruct parts of a surface that are not explicitly defined by the user. For example, in Fig. 2, portions of the surface that were not drawn explicitly by the user can be reconstructed. This ability to “predict” the surface intention and make it visible to the user is one of the primary advantages over other surface-based systems (Section 2.3) which require the user to manually define all observable portions of a surface.

While converting from points to surfaces is well understood through regression, being able to take a surface and “un-project” points off the surface is not as simple. If the user decides to modify a surface directly, without first modifying the points, then the updated surface will no longer match the points that it was originally fit to. Without any additional information, it is not possible to adjust the original points given only the fitted surface. The obvious solution would be to randomly distribute points above and below the surface according to some error criterion. However, fitting a surface to the new points is not guaranteed to reconstruct the surface exactly, thus breaking the representation correspondence.

To circumvent this issue, we introduce the concept of *residual matching*, whereby the prior residuals of the original points are used to recreate the residuals on a new surface. While this process is shown graphically for 2D shapes in Fig. 3 to simplify presentation, the technique is easily extended to three dimensions. When an WLS regression is fitted, the residuals of each of the fitted points is calculated and stored. Given a fitted surface, these residuals represent a signed distance from the surface to the original points. There is only one such surface which can uniquely minimize the square of these distances. When the old surface is modified, we can shift the original point positions such that the residuals with respect to the new surface are identical to those of the old surface. What this means is that fitting a new WLS function

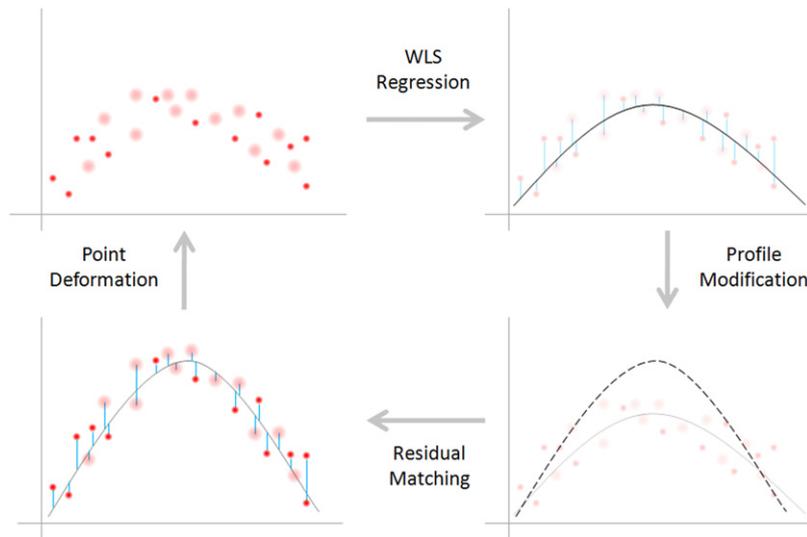


Fig. 3. A series of regressions and deformations is used to transition between the point representation (left side) and the surface representation (right side).

will result in an identical surface to the one that the adjusted points were “residual matched” around.

The *residual matching* approach allows our system to move fluidly between representations and allows interactive modifications on both representations, while maintaining direct correspondence between the two. An important limitation of this approach is that it requires a surface function that can be uniquely regressed. For deformable mesh surfaces, where there exists no functional form, the link between points and surfaces cannot be solved using this approach. In this regard, we provide a bi-directional link specifically between point-based and surface-based representations and, in the future, new techniques will have to be developed to bridge alternate sets of representations.

The effort required to maintain correspondence between the original point set and the fitted surface has several benefits over common alternatives, such as sampling new points directly from the fitted surface. First, the original point set reflects the initial uncertainty of the designer when creating a surface, and encodes useful information for designers if they ever decide to reevaluate their design choices. If they do not like the initial surface fit, they can return to their original point set to see how their original ideas deviated from the surface fitted by the regression. If we had simply sampled new points directly from the fitted surface, then we would be implicitly deciding on behalf of the designer that the uncertainty expressed in the original point set is unimportant. Second, in some of the techniques described in the next section, the user has the ability to modify the point representation, through various “pushing” or “pulling” gestures. If a new point set sampled from the surface were used in place of the original point set, the number and density of the points would always change, and the behavior of point modifications would be inconsistent and hard to control. By providing the correspondence with the original point set, we are always allowing the designer to directly modify his/her own input.

### 3.3. Interaction techniques

Once a set of multiple representations has been defined, the designer now requires adequate tools to manipulate and move between those representations. This section presents an overview of the AR system we use to demonstrate our techniques, then reviews the operations responsible for creating and modifying both point-based and surface-based shape representations. In addition, this section overviews interface techniques for trimming the resultant surfaces.

#### 3.3.1. AR system overview

When designing a virtual reality system, the choice of both input and output devices is important in determining the end-user functionality. This choice is non-trivial and there are several options, such as haptic devices and magnetically tracked pens, which all provide unique advantages for use in a product design system [44]. While it is not the focus of this paper, this section reviews some of the technology required to create the AR environment used in our system. In the case of a device for 3D sketching and surfacing, the following attributes were the primary design factors.

- It should have a natural, intuitive, hand-based interface for modeling/gesturing.
- There should be the provision of a range of input values in order to capture the flexibility of sketching (i.e. pen pressure).
- It should have the ability to operate at multiple size-scales, depending on the product being designed.
- It should preferentially have low cost, with the potential for multi-user interaction.

To provide this functionality, we used an optical tracking system to track a custom-made data input glove. The optical tracking system requires only consumer-grade webcams (Logitech 9000 Pros) to operate, and is based on OpenCV [45]. The glove, as shown in Fig. 4, has colored fingertips that can be optically tracked. In addition, it contains a pressure-sensitive resistor, a set of contact pads to detect hand gestures, and a small potentiometer for additional analog input.

In order to display the augmented world to the user, our system uses a head-mounted display (eMagin Z800 3Dvisor) with a monocular camera to provide augmented reality via video overlay. To perform head-tracking, we utilize the ARToolkitPlus tracking library [46]. Our system is able to achieve interactive rates (30 frames per second during normal operation) on modern computing hardware (Intel Core i7 2.93 GHz Desktop, 4.00 GB RAM, 1 GB NVIDIA GeForce GTS240 GPU). Optical occlusion can be mitigated using additional cameras.

Before describing the interaction techniques, it is first useful to understand the scope of different gestures or events which can trigger certain actions. Due to the contact pads in the glove, and the fact that each finger can be sensed independently, there are essentially four main gestures that are used by the system. Each is shown in Fig. 5(a)–(d). The “open” gesture (a) is used for point creation, the “closed” gesture (b) is used for point modification, the

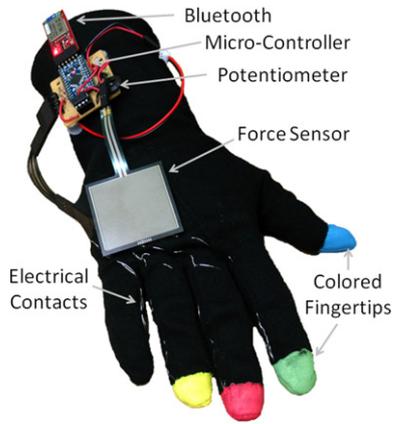


Fig. 4. Overview of the glove-based interface.

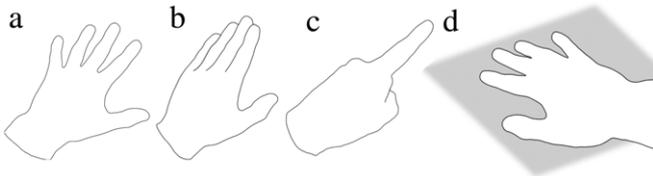


Fig. 5. The four primary gesture states: (a) open, (b) closed, (c) pointing, and (d) on table.

“pointing” gesture (c) is used for surface profile modification and trimming, and placing the hand on the table (d) activates a series of gesture commands for mode switching or surface rendering adjustments.

### 3.3.2. Point creation and modification

In order to create points, users can leave their hand in an open position and pass their hand through space, as if lightly gesturing the surface intention in the air. As users press on the glove’s force sensor, it creates points along the path of their hand whose size and transparency are proportional to the applied pressure. During point addition, the user has the option of also defining a symmetry plane and creating two sets of points that are symmetric about an arbitrary geometric plane. At this point, users can create additional points, or choose to deform the points already created.

Our point modification scheme is based on moving individual points in 3D space using the position and orientation of the user’s hand. By “pushing” individual sets of points, the user is able to manipulate their positions. However, only treating point movements individually can be time consuming and can create discontinuities in the surface. To combat this, we allow the user to propagate a portion of the movements to points other than those that are physically contacted. In this way, the point cloud resembles more an interconnected set of particles than a set of individual points.

If users choose to deform the preexisting points, they simply close their hand (Fig. 5(b)) and a circular modification disk is created. The disk’s center and radius are determined by the centroid of all the visible fingers, and the amount of pressure applied to the force sensor, respectively. To calculate the normal of the disk, vectors are created from the centroid to each of the fingers. The pairwise cross-products of these vectors are then averaged in order to find a representative normal that is roughly perpendicular to the palm. An example of this modification disk is shown in Fig. 6(b).

As this modification disk collides with points, the affected points are pushed in the normal direction of the plane. The disk is designed to only push the points. Hence, pulling would be achieved

by pushing the points from the opposite side. Depending on the motion of the modification disk, each affected point attains a unique displacement vector,  $v_i$ , that is parallel to the disk normal. If each point were independent, then only those points contacted by the plane would be moved. However, in order to provide a more meaningful deformation, these displacements are propagated to other nearby points in the field. For each contacted point  $i$ , a pairwise distance,  $d_{ij}$ , is calculated between itself and each non-contacted point  $j$  (Fig. 6(a)). The largest of these calculated pairwise distances is  $d_{max}$ . In order to control the extent to which these points deform other points, the user adjusts a potentiometer on the glove to a fraction between 0 and 1. Any point whose pairwise distance is less than this fraction times  $d_{max}$  lies within what we call the “propagation range” ( $d_{range}$  in Fig. 6(a)). For each point within the propagation range, its pairwise distance is then normalized with the propagation range ( $d_{ij}/d_{range}$ ). This normalized pairwise distance is then converted into an influence coefficient,  $\eta_{ij}$ , by using a falloff profile:

$$\eta(x) = \left( \exp\left(-\frac{x^2}{\sigma^2}\right) - \beta \right) / (1 - \beta), \quad (4)$$

where  $\beta = \exp(-1/\sigma^2)$  and  $x = d_{ij}/d_{range}$ . Here, the parameter  $\sigma$  is empirically determined to be  $1/\sqrt{2}$ . For each point in the propagation range, the total displacement vector is then calculated as

$$v_{j,\text{total}} = \frac{1}{m} \sum_{i=1}^m \eta_{ij} v_i, \quad (5)$$

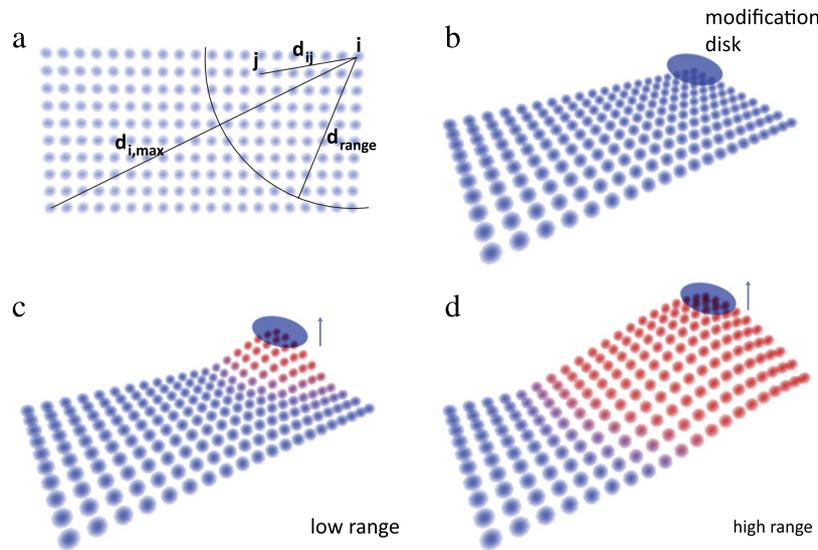
where the  $j$ th point is under the influence of  $m$  points. An example case is shown in Fig. 6, where the point at the top right is modified, and the modification propagates to other points.

By changing the extent to which individual points can affect one another, it is possible to make each point move and act independently or to make the entire point set respond together. Note that users can adjust the extent of the propagation range using the potentiometer attached to the glove, thus allowing them to control the stiffness of the deformation as desired. The idea of using deformation to propagate changes to nearby points has been explored before, including those based on spring–mass systems [47], point sets [36], and NURBS formulations [39]. Our method is similar in spirit to the point set deformation methods as no implicit topological connectivity is assumed. We have found our proposed method to be simple to implement yet to work suitably well, while recognizing that other approaches could be similarly adopted.

### 3.3.3. Surface creation and modification

Once the points have been created and modified, if desired, it is possible to construct a surface through the WLS approach defined in Section 3.2. For this, we first establish a local coordinate frame for the point set using principal component analysis (PCA). Here, the independent variables in the polynomial regression ( $x$  and  $y$  in Eq. (2)) are aligned in the directions of maximum variance of the point cloud. The residuals within the WLS regression are calculated as the perpendicular distances to the base plane. This PCA base-plane recalculation is done whenever a user chooses to fit a new surface to the point set; unless the user chooses to refit the plane, any additional points are measured with respect to the original base plane. It should be noted that we are aiming to interpret point sets as open, 2-manifold surfaces of genus 0.

Once the surface has been created, the user also has the option of modifying the surface directly using *profile modifications*. The user can correct certain profiles in the surface by sketching a stroke representing the modification. This is done by making the



**Fig. 6.** Point modification. The modifications are propagated through the point set using pairwise distances. Given our distance nomenclature (a), an initial 3D point set is approached by the modification disk (b). An input on the glove specifies to what extent the changes are propagated (c–d).

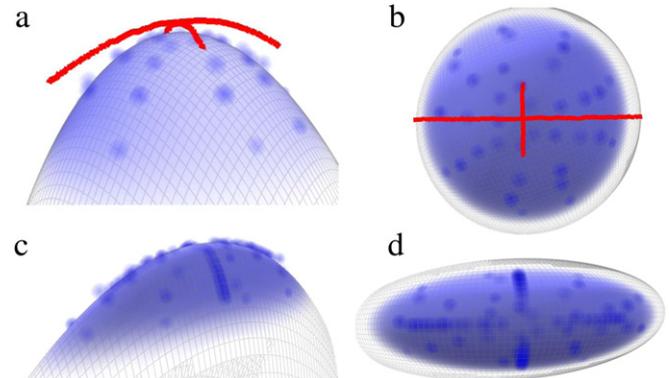
“pointing” gesture, and applying pressure to the force sensor. Once a freeform 3D curve has been created, it is smoothed using a Savitzky–Golay filter [48] and evenly resampled. Now the challenge becomes how to modify a surface given an arbitrary curve in 3D space. Since the single sketched curve is not necessarily quadratic, the surface cannot be interpolated directly from the curve using standard techniques.

In order to update the surface in a meaningful way, given a profile modification curve, we again take advantage of the dual representation. The surface is temporarily reverted back into the point representation (as described in Section 3.2), and the new points from the modification curve are added as temporary points such that they will affect the WLS regression. By treating a curve as a set of points, the user is not limited to single-stroke curve modifications or region of interest definitions as seen in prior sketch-based modification techniques [49,50]. The user can freely define multiple, oversketched curves, and our approach can process all of them simultaneously. An example of this can be seen in Fig. 7(c)–(d).

However, before any modification curve can be incorporated into the point set for regression, two non-trivial attributes must be determined: (1) how many of the points along a new curve should be used in the regression, and (2) what pressures should these points be assigned. Both of these parameters will affect how much of an effect a modification curve will have on the regression, and thus they need to be chosen carefully.

In order to adaptively decide on how finely the modification curve should be resampled, we use the current size of the point cloud as a guide. If the point cloud is only 10 points in size, then incorporating a curve with 1000 resampled points would drown out any contributions from the original point set. To mitigate this problem, we resample the curve such that the number of new curve points is equal to the size of the original point set. The justification for this is that, if the curve did not grow at a rate proportional to the point set, then the effect of a modification curve would eventually become negligible compared to the point set. These points are only temporarily included for the purpose of the modification, and are deleted once the modification is complete.

Even if the number of points on the modification curve is chosen appropriately, the choice of the artificially assigned pressures can have a large impact on how the surface is modified. The higher the assigned pressure, the closer the surface will match the modification curve. For this purpose, we assign the modification



**Fig. 7.** Given an original surface and point set, the user is able to use a sketched modification curve to adjust the surface (a, b). The points move such that their new residuals are identical to the residuals prior to the adjustment. This method can handle overstroking and sketching over multiple parts of the surface simultaneously (a, b). The resulting surface (c, d) is adjusted using contributions from all curves, irrespective of spatial or temporal order.

curve with the maximum possible pressure capable of being measured by the force sensor ( $\rho = 1$ ). This choice gives the modification curve a large degree of influence over the surface modification, while still keeping its influence within the range of user-specified pressures.

Once these new temporary points are factored in, a new WLS regression surface is fit and the original points are updated using our residual matching approach. However, what if the user wants to include some of the modification curve points in the point representation? The challenge now becomes how to correctly factor in the new curve into the point set such that the surface/point correspondence is maintained. If the curve points were simply added “as is”, then not only would we be doubling the size of the point set, but we would also be introducing points with above average pressure values. Moreover, since the new points were not part of the original regression, we cannot “residual match” them in a way that preserves the surface correspondence.

In order to incorporate new curve points while maintaining surface/point correspondence, we take advantage of the fact that any points that lie exactly on the fitted surface will have no contribution to the final WLS residual. If the fitted surface already represents the function which minimizes the overall residuals,

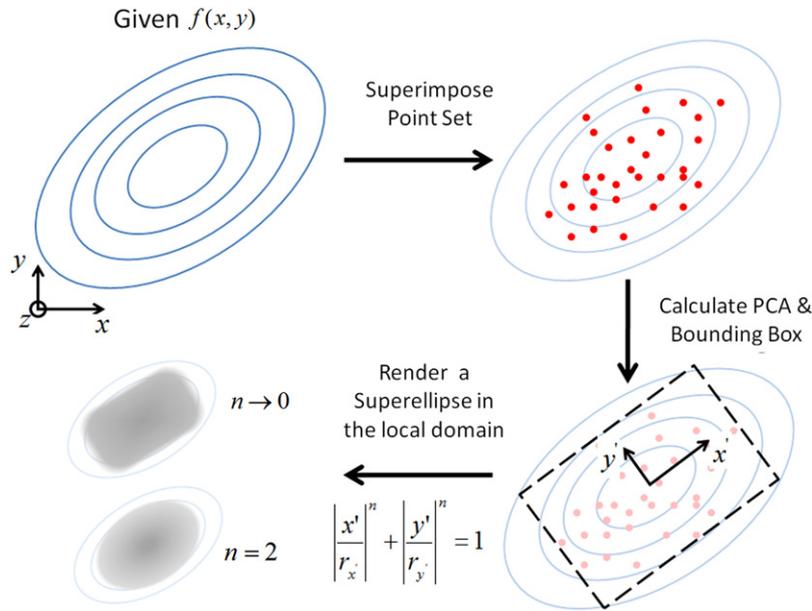


Fig. 8. Users can change the rendered surface shape using a superellipse.

then placing new points directly on that surface will maintain that minimum. As a result, by projecting the original curve points onto the surface, we can freely add additional points while still maintaining the surface to point correspondence. To prevent doubling the size of the point set, we resample the curve such that the new curve only contributes 10% of the original point set size. For example, given a point set size of 100, a new profile modification would contribute 10 points, increasing the total size to 110 points. To resolve the problem of introducing a large number of high-pressure points, we instead assign the new points a pressure that is equal to the average pressure in the original point set. This combination of point size and pressure was found empirically to be sufficient at capturing the users' intentions, while still allowing drastic changes in the surface, if desired.

An example of a surface modification is shown in Fig. 7. In Fig. 7(a)–(b), a modification curve is used to adjust the surface and preexisting points, resulting in those seen in (c) and (d). The residuals between the points and the underlying surface remain the same. This demonstrates how a set of multiple curves can adjust different parts of the surface. A user can also manually partition a stroke and use the same procedure to update multiple surfaces simultaneously.

### 3.3.4. Surface trimming and rendering

The last critical interaction step is how surfaces are trimmed and subsequently rendered. A central problem with the use of functional surfaces is that the surface is not naturally bounded/trimmed. A strict functional surface representation does not implicitly define the bounds of the surface, and thus presents a challenge when displaying it to the user. Using a predefined boundary would solve this problem; however, it would severely limit users' abilities to define their own desired shapes. Ultimately the key challenge is to provide a way to specify the desired bounds of the surface in a way that is fast and easy, as well as high-fidelity enough for the user's needs.

To balance these needs, we provide two techniques: (1) define surface boundaries by the bounding box of the point representation and use a modifiable superellipse formulation to let the user adjust the shape, and (2) let the user sketch a boundary curve in 3D space and use its projection on the surface to define the boundary. The first method is fast and intuitive to use for quick surface

mockups, while the second one affords the user more detailed control over the surface bounds when needed. In our case the surface is rendered as a set of blended polygons in OpenGL, and as such when we refer to points on the surface we are referring to points that comprise the polygonal surface mesh.

*Superellipse-based visualization.* Our first approach to solving the visualization problem is to compute an optimally oriented bounding box using the point representation projected onto the on the PCA base plane, and then allow a user to render a superellipse within that bounding box. This process is shown graphically in Fig. 8. First, PCA is performed on the point representation of the surface to define a local 3D plane whose  $x'$  and  $y'$  are aligned with the two largest principal components. All points are then projected onto this 2D plane. The bounding box of all points is computed within in local coordinate system. From here, a superellipse is rendered, according to the following equation:

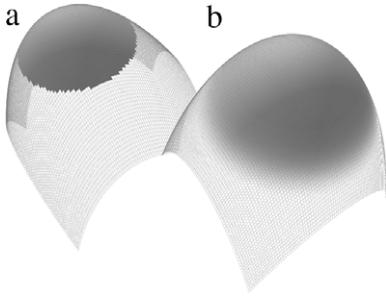
$$f(x', y') = \left| \frac{x'}{r_{x'}} \right|^n + \left| \frac{y'}{r_{y'}} \right|^n, \tag{6}$$

where  $f(x', y') = 1$  contains some points which lie at the extent of the bounding box. By allowing the designer to change  $n$  from  $0 \rightarrow \infty$ , the shape can change on a continuum from a square to an ellipse to a star shape. In order to give a less distinct end to the surface, the surface transparency  $T(x', y')$  is set such that

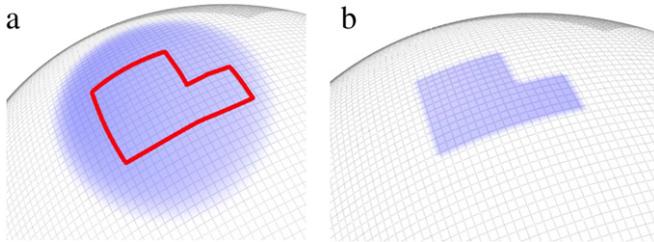
$$T(x', y') = \exp\left(\frac{-f(x', y')^6}{10}\right), \tag{7}$$

and is truncated between 0 and 1. This creates a smooth but quick transition from the opaque region to the transparent region. A comparison between this transparent falloff and no transparent falloff can be seen in Fig. 9.

*Boundary curve trimming.* Design situations may arise where the shapes created by the superellipse method will not be sufficient for the needs of the user, such as when the user wants to visualize parts of the surface outside the point bounding box. In these cases, we provide a way to sketch an arbitrary boundary curve to define the extent of the surface. To do this, the user simply draws an approximately closed curve in 3D space using the “pointing” gesture (Fig. 5(c)). However, standard trimming methods, such as



**Fig. 9.** Comparison of an identical surface rendered with (a) no transparency, and (b) transparency falloff (Eq. (7)). The underlying OpenGL quadrilateral mesh is overlaid only for presentation purposes.



**Fig. 10.** An original surface (a) can be trimmed with a freeform boundary curve such that the transparency falloff lies along that boundary (b). The underlying surface mesh is overlaid purely for presentation purposes.

user-perspective ray-casting approaches or nearest-distance-to-surface mappings, produce results that are either jagged or not well aligned with what the user had anticipated.

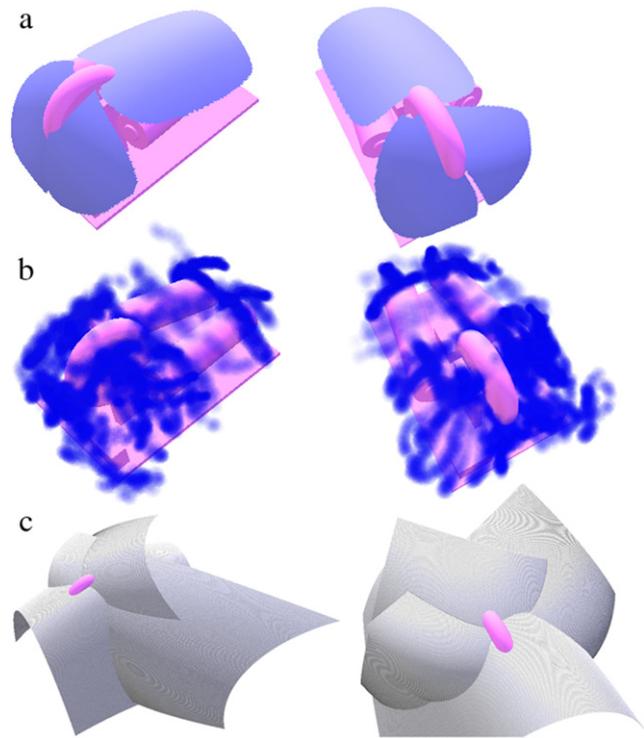
To resolve this issue, the boundary curve is projected onto a PCA plane fitted to the point representation, as described earlier. Once all points are projected in two dimensions, each point on the surface mesh is assigned a transparency value based on its distance to the closed projected boundary curve. This distance is calculated as the Euclidean distance from the point on the surface to the closest sampled point on the boundary curve. Then a point-in-polygon ray-casting algorithm is used to calculate whether a point is inside or outside the bounding polygon, and points outside receive a negative distance. The transparency for each point is then calculated using a logistic function:

$$T_i = \frac{1}{1 + e^{-10 \frac{d_i}{d_{\max}}}}, \quad (8)$$

where  $d_{\max}$  is the maximum observed point distance. This produces a nicely blended surface boundary along an arbitrary path. In this manner, surface trimming is performed without the need to actually modify the mesh structure, and with results that more robustly mirror the user's expectations. An example of the resulting trimmed surface is shown in Fig. 10.

#### 4. Results

In order to demonstrate the methods presented in this paper, we have applied them to several design tasks. Fig. 11 shows a set of three surfaces representing a scene from the exterior conceptual design process of a mouse. Fig. 11(a) shows the trimmed surfaces, whereas (b) illustrates the underlying point representation. Fig. 11(c) displays a mesh representation derived from the surface sets created in (a). This demonstrates that the surface can be analytically extended across larger parts of the domain. At any stage of the conceptual design exploration, the user is able to export these “infinite” surfaces and perform more detailed operations on them via third-party integrated CAD



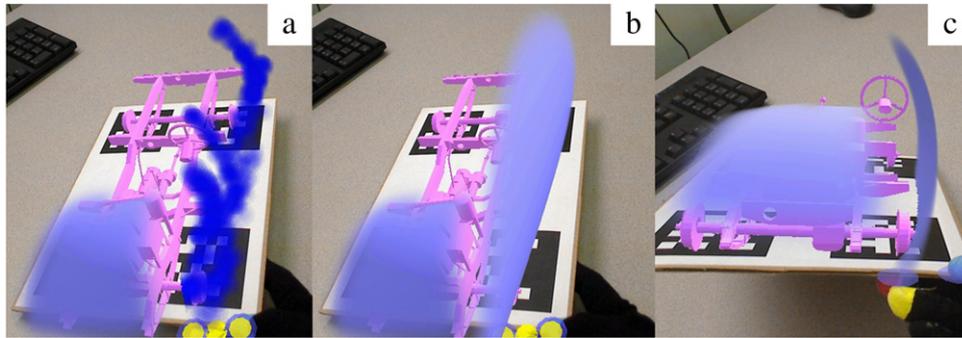
**Fig. 11.** A set of computer mouse external surfaces designed around a preexisting geometric constraint. The figure shows (a) the bounded surface representation, (b) the point set representation, and (c) the underlying surface mesh.

software. We illustrate this functionality in some of the examples presented in this section in order to demonstrate how our methods fit into a larger design process.

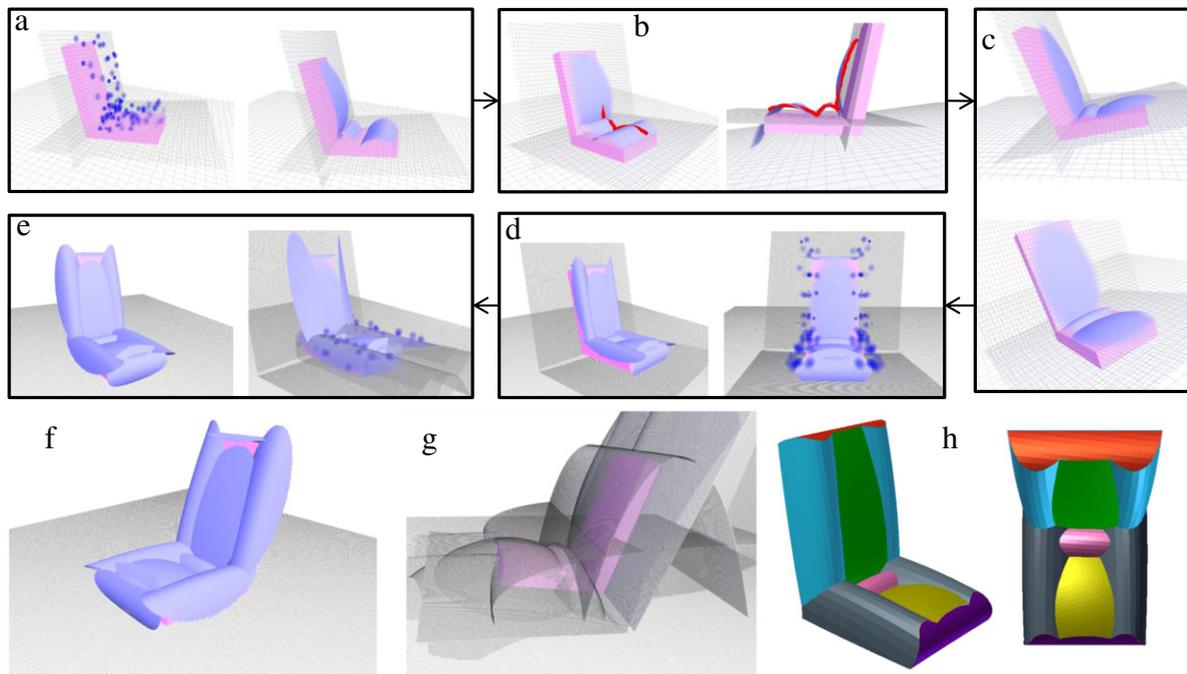
Fig. 12 presents the view of the user through the Head-mounted display (HMD) during modeling operations of parts of a car chassis. Here, the effect of our surface fitting approach is shown in one of the exterior side surfaces.

In order to demonstrate the design implications of multiple shape representations, Fig. 13 shows an example progression of the user while designing a car seat. Fig. 13(a) shows the initial underlying geometry, in this case the internal structure of a car seat and the three user-created primary surfaces. The user designed these surfaces sequentially by adding points and fitting a quadratic WLS surface. Using the surface representation, the user then sketches a single modification stroke and adjusts the three surfaces simultaneously to increase the curvature (c). Note that these surfaces are symmetric. The user then decides to bound these surfaces with additional side features. These surfaces are also created utilizing symmetry to result in identical surfaces at both sides of the car seat (d). The user then decides to add additional curvature to these side rests. To do this, the user switches back to the point representation and applies point modification to the model (e) to better define the desired curvature. This results in a difference in curvature in the side rests between (d) and (e). Finally, the user creates the top and bottom bounding surfaces (d–e) and finalizes the design (f) through additional surface modification and point modification. Both the point representation and the untrimmed, analytical surfaces remain intact throughout all operations. Note that the underlying mesh structure (g) for each of the designed surfaces can be extended beyond those areas strictly sketched by the user. These surfaces can then be exported to third-party CAD software for further refinement and trimming operations (h), which are beyond the scope of the current work.

To further exemplify the capabilities enabled by our techniques, part of an external mouse design conceptual exploration process



**Fig. 12.** An example of a car surface being designed. (a) shows the point cloud laid down by the user, and (b) and (c) show the resulting surface.



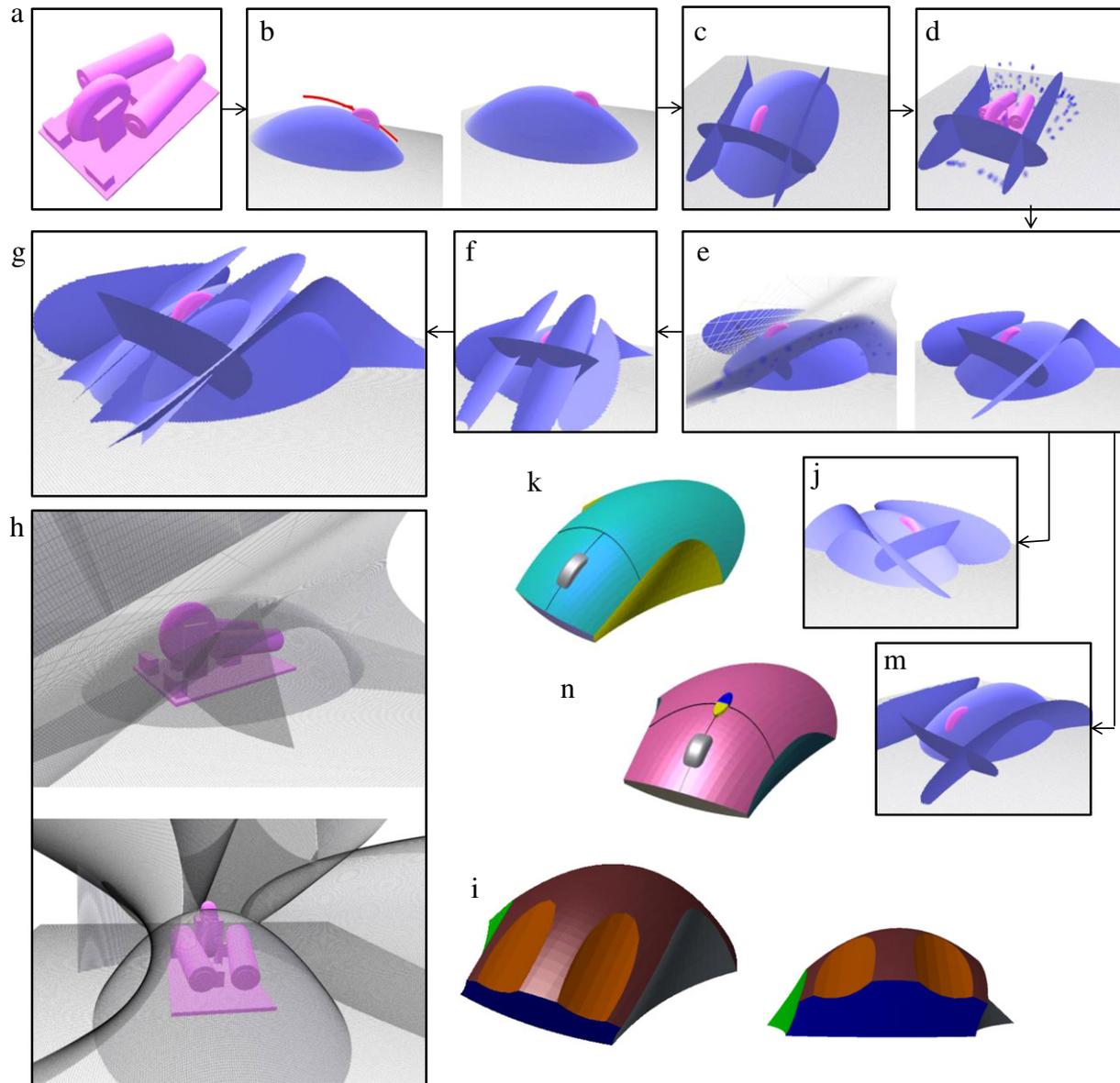
**Fig. 13.** An example of a user progression through multiple surface representations. Starting by creating three surfaces (a), the user progresses to the final design (f–g–h) through a series of operations that utilize both the point and surface representations. (a) demonstrates the initial point sets created by the user and the resulting quadratic WLS fitted surfaces, all of which utilize symmetry. In (b), the user sketches a single 3D stroke and this stroke is used to update the surfaces simultaneously (c). In (d), the user adds new points and creates additional symmetric surface sets, and then, going back to the point representation, the user modifies the surfaces further by adding points (e). (f) and (g) show the final surfaces designed and the underlying mesh structure (which can be infinitely extended), respectively. (h) is the result after the surfaces are trimmed via integrated commercial CAD software.

is presented in Fig. 14. Fig. 14(a) represents the possible layout of the internal structure, around which external mouse surfaces are to be designed. The user first lays out an initial set of points and creates a primary surface through WLS fitting (b). Realizing that the initial surface was passing through the internal structure, the user then modifies this surface by adding a 3D modifier stroke (b). The user then creates three secondary feature surfaces to be incorporated in the mouse design by adding additional points and fitting operations (c). The user decides to increase the curvature of the primary surface, and changes to the point representation to add new points to the corresponding point set (d). Our point modification scheme is utilized to give the desired shape to all of the three feature surfaces (e). As shown in the side features, the user can deactivate symmetry constraints if desired. The user creates additional symmetric detail surfaces, and translates these additional surfaces onto the design object to finalize the design (g). The underlying infinite surface representations are then exported by the user to a commercial CAD system for further trimming operations (i). Note that, throughout the design process, the user explores several other designs, two of which are illustrated in

Fig. 14. The user first modifies the primary surface to further increase the curvature, and then applies symmetry to the left side feature surface (j–k). Another design evaluated by the user is shown in Fig. 14 (m–n), where the user decides to apply symmetry to the right side feature, discarding the original left side one, and notably decreasing the curvature of the primary surface.

## 5. Discussion

*On the use of multiple shape representations and moving between them, and the utility of visualizing “uncertainty”.* Our system presents the use of both point-based and surface-based representations for the purpose of exploring conceptual designs without the need for detailed surface operations such as trimming or continuity enforcement in early design stages. The modeling process under a point representation is noticeably different than under a surface representation. Point-based modeling, given the transparent and variable nature of the rendering technique, is initially used to rapidly express the overall form of the surface and then later used to allow the designer to explore alternative design ideas. By



**Fig. 14.** An example of conceptual design exploration. Starting from the underlying geometry (a), the user creates a symmetric primary surface and modifies it with a single 3D stroke using our profile modification scheme. The user then adds three secondary surfaces that constitute the secondary features of the mouse (c). Going back to the point representation, the user adds points to modify the primary surface (d). Utilizing several steps of point addition and modification, the user shapes secondary features (e). At this step, the user adds additional detail surfaces (f). These are created utilizing symmetry well above the design object, then placed correctly through translation (g). The final design (g) and the underlying infinite surface representations (h) are then trimmed via integrated commercial CAD software (i). Note that, after step (e), the user explores the design space by modifying the primary surface through point modification. Two different exploration paths are shown in steps (j–k) and steps (m–n), respectively. This demonstrates how this system can be used for storing and exploring different design alternatives.

contrast, the surface representation allows the designer to get concrete feedback and visualization of the current design, and to perform more detailed surface adjustments.

This paper proposes a technique based on WLS regression in order to provide a bi-directional correspondence between point-based and surface-based representations. The choice of the two representations, points and analytical surfaces, is guided by two observations. (1) Points are a powerful yet flexible data structure whose modification techniques can become sophisticated if needed, and (2) the final surfaces in product designs are typically parametric surfaces which are a subset of the analytical surfaces upon which our approach is based. This paper focuses on the new tools and techniques required to support multiple representations, rather than an in-depth comparison of all possible alternative representations. As such, mesh-based deformation techniques were not explored by our system. The solution of the weighted

least squares regression approach requires the solution of a set of  $N$  linear equations ( $N = \#$  of coefficients in Eq. (2)). This results in the decomposition of an  $N$  by  $N$  matrix, which becomes more computationally expensive as the order of the polynomial equation gets higher. For consumer product design applications, where the surfaces tend to be cubic or less, this regression can be done at interactive rates, even with large point clouds. In the examples shown in this paper, the surface fitting took no longer than a few seconds on modern computing hardware.

*On intuitive methods for creating, and deforming these multiple representations.* Our point deformation techniques provide the user with a way to deform a range of points using the motions of only a few points. We allow the user to adjust this effect during operation. More advanced particle techniques (Section 2) utilize force-based interactions to create deformations that mirror clay or other physical systems more closely. Our system does not attempt

this level of complexity, opting instead for a faster, geometry-based method. This trade-off was found to be acceptable for adjusting the general shape of the point cloud, given the preexisting uncertainty of the points. If this deformation technique were used on a more concrete, deformable mesh representation, it may need to be revisited or enhanced.

Profile modification allows the user to adjust the current surface to a given 3D stroke using regression. Since this approach uses only resampled stroke points, it is inherently invariant to stroke direction and order. This means that the same approach can handle oversketching or even sketching separate parts of a surface. While the implementation of the technique appears similar to that of regular surface fitting, it has a few additional limitations that arise. First, since each surface is based on residuals calculated specifically with respect to its own local PCA plane, residual matching requires that the normal of the plane remain constant. The implication of this is that users cannot attempt to drastically alter the general orientation of the surface using a single stroke, as this will lead to an unexpected fit. This is not typically an issue, as surface modifications using this method are used mostly for minor surface adjustments whose overall PCA directions remain similar. We also have a multiple surface profile modification scheme, demonstrated in Fig. 13 (b)–(c), which enables users to modify multiple surfaces with a single 3D modifier stroke. We achieve this by allowing the user to partition a single stroke and then employ our standard profile modification method on each of the surfaces.

*On representing various geometric entities for aiding conceptual design.* It was also postulated that the inclusion of a “fuzzier” representation would help designers visualize the uncertainty related to the conceptual design. While the rendering of the points using a variable transparency provided this “fuzzier” appearance, it is not the only method for displaying such variability. Other potential methods of visualization include adding small points in different densities and patterns or potentially using vertices in surface meshes as points. While we explored these methods, the visualization approach shown in Fig. 2 demonstrates some of the advantages that this choice of rendering has over more concrete mesh representations. In addition, the surface boundaries are rendered with transparency such that their extent also maintains some incompleteness.

Surface trimming is provided as a means to control how much of the surface is displayed to the user. Our boundary trimming is accomplished by the following two methods: a superellipse-defined zone using the bounding box of the point set, and a closed-loop 3D curve projected onto the surface domain. Our first approach, using a superellipse to define the boundary, cannot provide the same level of detail as the freeform curve approach. However, it often suffices as a quick method of defining the surface boundary due to the following reasons: (1) the extent of the point bounding box is a good indicator as to the extent of the desired surface, and (2) the shapes afforded by the superellipse (square, ellipse, etc.) are often good approximations of the user’s final intended shape. This automatic surface definition is valuable in early conceptual design, where users do not want to spend much of their time accurately defining the surface boundaries.

## 6. Summary

This paper presents an AR product design system that utilizes multiple shape representations for the construction of freeform surfaces. The primary contributions of the work are (1) the formulation of virtual shape data in multiple, simultaneous representations (points and surfaces), and a regression based method to transition fluidly between these representations during design, (2) intuitive methods for deforming and exploring product shape using these multiple representations, and (3) representing these

forms such that designers can explore conceptual designs without the need for detailed surface operations such as trimming or continuity enforcement. The shape representations explored by this work are 3D point sets and analytical surfaces. By maintaining a correspondence between point-based and surface-based representations, we demonstrate how multiple representations can provide additional modeling capabilities beyond the sum of their parts.

The computational system uses a glove-based interface and a head-mounted display to create an immersive styling environment for the user. Within this environment, the user can interactively create and modify point sets and surfaces. User-controlled pressure values are used to render the point set representation with different degrees of perceptual ambiguity. Analytical surfaces are then fit to point sets using WLS regression. An approach called *residual matching* is introduced as a means of maintaining correspondence between points sets and analytical surfaces during deformations.

In order to modify point-based and surface-based representations, a set of techniques is presented: (1) point deformation by point movement propagation, (2) surface modification through sketched profiles, (3) switching between representations using a passive-haptics gesture system, and (4) surface trimming using a boundary-controlled transparency function. The methods are presented and discussed in the context of freeform surface design, with examples of a computer mouse external enclosure and the design of a car seat.

By combining the use of point set and surface-based shape representations, our system allows prior interactive modeling techniques to be combined and reused in novel ways. It allows designers to utilize these techniques in ways that they are most comfortable with, instead of requiring adherence to a single set of representations. While this work provides only a small set of such modeling operations, the multiple representation framework we provide is broad enough to encompass a variety of new or existing techniques. By providing this perspective, we hope that future generations of computer-aided conceptual design technologies can promote design creation and exploration in ways not previously possible with single shape representations.

## Acknowledgments

The authors would like to acknowledge the editor and the four anonymous reviewers for their helpful comments and suggestions. Their efforts have significantly improved the quality of this manuscript.

## References

- [1] Horváth I. On some crucial issues of computer support of conceptual design. In: Product engineering. Netherlands: Springer; 2005. p. 123–42.
- [2] Yamaguchi Y, Nakamura H, Kimura F. Probabilistic solid modeling: a new approach for handling uncertain shapes. In: Selected and expanded papers from the IFIP TC5/WG5.2 working conference on geometric modeling for product realization. Amsterdam (The Netherlands, The Netherlands): North-Holland Publishing Co.; 1992. p. 95–108.
- [3] Jin Y, Benami O. Creative patterns and stimulation in conceptual design. In: AI EDAM 24. 2010. p. 191–209 [Special Issue 02].
- [4] Benami O, Jin Y. Creative stimulation in conceptual design. In: ASME 2006 international design engineering technical conferences. 2002.
- [5] Chrysikou EG, Weisberg RW. Following the wrong footsteps: fixation effects of pictorial examples in a design problem-solving task. *Journal of Experimental Psychology* 2005;31(5):1134–48.
- [6] Perttula M, Liikkanen L. Structural tendencies and exposure effects in design idea generation. In: ASME 2006 international design engineering technical conferences. 2006.
- [7] Tseng I, Moss J, Cagan J, Kotovsky K. The role of timing and analogical similarity in the stimulation of idea generation in design. *Design Studies* 2008;29(3): 203–21.
- [8] Hsu W, Liu B. Editorial—conceptual design: issues and challenges. *Computer-Aided Design* 2000;32:849–50.

- [9] Horváth I, Rusák Z, Vergeest J, Kuczogí G. Vague modeling for conceptual design. In: TMCE 2000 conference. 2000. p. 131–44.
- [10] Wang L, Shen W, Xie H, Neelamkavil J, Pardasani A. Collaborative conceptual design—state of the art and future trends. *Computer-Aided Design* 2002; 34(13):981–96.
- [11] Robertson B, Radcliffe D. Impact of cad tools on creative problem solving in engineering design. *Computer-Aided Design* 2009;41(3):136–46. Computer support for conceptual design.
- [12] Zeng Y, Pardasani A, Dickinson J, Li Z, Antunes H, Gupta V, et al. Mathematical foundation for modeling conceptual design sketches. *Journal of Computing and Information Science in Engineering* 2004;4(2):150–9.
- [13] Cole F, Sanik K, DeCarlo D, Finkelstein A, Funkhouser T, Rusinkiewicz S. et al. How well do line drawings depict shape? In: *ACM transactions on graphics*. Proc. SIGGRAPH. vol. 28. 2009.
- [14] Nealen A, Igarashi T, Sorkine O, Alexa M. Fibremesh: designing freeform surfaces with 3D curves. In: *ACM SIGGRAPH 2007*. ACM transactions on computer graphics. 2007.
- [15] Masry M, Kang D, Lipson H. A freehand sketching interface for progressive construction of 3D objects. *Computer Graphics* 2005;29:563–75.
- [16] Kara LB, Shimada K. Sketch-based 3D-shape creation for industrial styling design. *IEEE Computer Graphics and Applications* 2007;27(1):60–71.
- [17] Igarashi T, Matsuoka S, Tanaka H. Teddy: a sketching interface for 3D freeform design. In: *ACM SIGGRAPH 99*. 1999.
- [18] Wesche G, Seidel H-P. Freedrawer: a free-form sketching system on the responsive workbench. In: *Proceedings of the ACM symposium on virtual reality software and technology*. VRST '01. 2001. p. 167–74.
- [19] Keefe DF, Feliz DA, Moscovich T, Laidlaw DH, LaViola Jr JJ. Cavepainting: a fully immersive 3D artistic medium and interactive experience. In: *13D 01: proceedings of the 2001 symposium on interactive 3D graphics*. New York (NY, USA): ACM; 2001. p. 85–93.
- [20] Koutamanis A. Fuzzy modelling for early architectural design. *International Journal of Architectural Computing* 2007;5(4):589–610.
- [21] Lodwick WA, Santos J. Constructing consistent fuzzy surfaces from fuzzy data. *Fuzzy Sets and Systems* 2003;135(2):259–77.
- [22] Martin RR. Inexact shape modelling with fuzzy sets. In: *Proceedings of the set-theoretic solid modelling: techniques and applications conference*. CSG 94. 1994. p. 73–98.
- [23] Blinn JF. A generalization of algebraic surface drawing. *ACM Transactions on Graphics* 1982;1(3):235–56.
- [24] Rusák Z, Horváth I. *Product engineering*. Springer Netherlands; 2005. [Chapter] A new geometry representation for multipurpose modeling in conceptual design.
- [25] Lim SW, Lee BS, Duffy AHB. Incremental modelling of ambiguous geometric ideas (i-magi): representation and maintenance of vague geometry. *Artificial Intelligence in Engineering* 2001;15(2):93–108.
- [26] Hoppe H, Derosé T, Duchamp T, McDonald JA, Stuetzle W. Surface reconstruction from unorganized points. *Computer Graphics* 1992;26(2):71–8.
- [27] Vieira M, Shimada K. Surface mesh segmentation and smooth surface extraction through region growing. *Computer Aided Geometric Design* 2005; 22(8):771–92.
- [28] Lielieveldt BPF, der Geest RJV, Rezaee MR, Bosch JG, Reiber JHC. Anatomical model matching with fuzzy implicit surfaces for segmentation of thoracic volume scans. *IEEE Transactions on Medical Imaging* 1999;8(3).
- [29] Sachs E, Roberts A, Stoops D. 3-Draw: a tool for designing 3D shapes. *IEEE Computer Graphics and Applications* 1991;11(6):18–26.
- [30] Jimeno A, Puerta A. State of the art of the virtual reality applied to design and manufacturing processes. *The International Journal of Advanced Manufacturing Technology* 2007;33(9):866–74.
- [31] Grossman T, Balakrishnan R, Kurtenbach G, Fitzmaurice G, Khan A, Buxton B. Creating principal 3D curves with digital tape drawing. In: *ACM CHI 2002 conference on human factors in computing systems*. 2002.
- [32] Ye J, Badiyani S, Raja V, Schlegel T. Applications of virtual reality in product design evaluation. In: *Human-computer interaction. HCI applications and services*. Lecture notes in computer science, vol. 130. Springer; 2007. p. 1190–9. [Chapter].
- [33] Willis KDD, Lin J, Mitani J, Igarashi T. Spatial sketch: bridging between movement & fabrication. In: *TEI'10: proceedings of the fourth international conference on tangible, embedded, and embodied interaction*. New York (NY, USA): ACM; 2010. p. 5–12.
- [34] Fiorentino ASM, De Amicis R, Monno G. Spacedesign: a mixed reality workspace for aesthetic industrial design. In: *ISMAR 2002 IEEE and ACM international symposium on mixed and augmented reality*. 2002.
- [35] Fleisch T, Brunetti G, Santos P, Stork A. Stroke-input methods for immersive styling environments. In: *Proceedings shape modeling applications*, 2004. IEEE; 2004. p. 275–83.
- [36] Dorman J, Rockwood A. Surface design using hand motion with smoothing. *Computer-Aided Design* 2001;33(5):389–402.
- [37] Keefe D, Zeleznik R, Laidlaw D. Drawing on air: input techniques for controlled 3D line illustration. *IEEE Transactions on Visualization and Computer Graphics* 2007;13(5):1067–81.
- [38] Anabuki M, Ishii H. Ar-jig: a handheld tangible user interface for modification of 3D digital form via 2D physical curve. In: *ISMAR'07: proceedings of the 2007 6th IEEE and ACM international symposium on mixed and augmented reality*. Washington (DC, USA): IEEE Computer Society; 2007. p. 1–10.
- [39] Perles BP, Vance JM. Interactive virtual tools for manipulating NURBS surfaces in a virtual environment. *Journal of Mechanical Design* 2002;124(2):158–63.
- [40] Rupérez M, Monserrat C, Alemany S, Juan M, Níz MA. Contact model, fit process and, foot animation for the virtual simulator of the footwear comfort. *Computer-Aided Design* 2010;42(5):425–31. Advanced and emerging virtual and augmented reality technologies in product design.
- [41] Schkolne S, Pruett M, Schröder P. Surface drawing: creating organic 3D shapes with the hand and tangible tools. In: *CHI'01: proceedings of the SIGCHI conference on human factors in computing systems*. New York (NY, USA): ACM; 2001. p. 261–8.
- [42] Chang Y-H, Chen Y-T, Chang C-W, Lin C-L. Development scheme of haptic-based system for interactive deformable simulation. *Computer-Aided Design* 2010;42(5):414–24. Advanced and emerging virtual and augmented reality technologies in product design.
- [43] Bourdot P, Convard T, Picon F, Ammi M, Touraine D, Vézien J-M. Vrcad integration: multimodal immersive interaction and advanced haptic paradigms for implicit edition of cad models. *Computer-Aided Design* 2010;42(5):445–61. Advanced and emerging virtual and augmented reality technologies in product design.
- [44] Bowman DA, Kruijff E, LaViola JJ, Poupyrev I. *3D user interfaces: theory and practice*. Redwood City (CA, USA): Addison Wesley Longman Publishing Co., Inc.; 2004.
- [45] Bradski G, Kaehler A. *Learning OpenCV: computer vision with the OpenCV library*. 1st ed. O'Reilly Media, Inc.; 2008.
- [46] Wagner D, Schmalstieg D. Artoolkitplus for pose tracking on mobile devices. In: M. Grabner, H. Grabner, (Eds.), *12th computer vision winter workshop*. CVWW'07. St. Lambrecht, Austria. 2007. p. 139–46.
- [47] Ix FD, Qin H, Kaufman A. A novel haptics-based interface and sculpting system for physics-based geometric design. *Computer-Aided Design* 2001; 33(5):403–20.
- [48] Savitzky A, Golay MJE. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry* 1964;36(8):1627–39.
- [49] Kara LB, Shimada K. Construction and modification of 3D geometry using a sketch-based interface. In: *Eurographics workshop on sketch-based interfaces and modeling*. 2006. p. 59–66.
- [50] Botsch M, Kobbelt L. An intuitive framework for real-time freeform modeling. In: *SIGGRAPH'04: ACM SIGGRAPH 2004 papers*. New York (NY, USA): ACM; 2004. p. 630–4.